



# A global Newton-type scheme based on a simplified Newton-type approach

Mario Amrein<sup>1</sup>

Received: 21 December 2019 / Revised: 9 April 2020 / Accepted: 30 June 2020

© The Author(s) 2020

## Abstract

Globalization concepts for Newton-type iteration schemes are widely used when solving nonlinear problems numerically. Most of these schemes are based on a predictor/corrector step size methodology with the aim of steering an initial guess to a zero of  $f$  without switching between different attractors. In doing so, one is typically able to reduce the chaotic behavior of the classical Newton-type iteration scheme. In this note we propose a globalization methodology for general Newton-type iteration concepts which changes into a simplified Newton iteration as soon as the transformed residual of the underlying function is small enough. Based on Banach's fixed-point theorem, we show that there exists a neighborhood around a suitable iterate  $x_n$  such that we can steer the iterates—without any adaptive step size control but using a simplified Newton-type iteration within this neighborhood—arbitrarily close to an exact zero of  $f$ . We further exemplify the theoretical result within a global Newton-type iteration procedure and discuss further an algorithmic realization. Our proposed scheme will be demonstrated on a low-dimensional example thereby emphasizing the advantage of this new solution procedure.

**Keywords** Global Newton methods · Simplified Newton method · A posteriori analysis · Newton path

**Mathematics Subject Classification** 37N30 · 46N40 · 58C15 · 65H10 · 49M15

## 1 Introduction

For the time being, let  $U \subset \mathbb{R}^n$  be open and  $f : U \rightarrow \mathbb{R}^n$  be of class  $C^1(U; \mathbb{R}^n)$ . In this note we are interested in finding the zeros  $x \in U$  of  $f$ , i.e., we aim to solve the equation

---

✉ Mario Amrein  
mario.amrein@zhaw.ch

<sup>1</sup> ZHAW, Technoparkstrasse 2, 8400 Winterthur, Switzerland

$$x \in U : f(x) = 0. \quad (1)$$

In general—apart from trivial toy problems—the solutions  $x_\infty$  can only be computed numerically. Here, we focus on the following approach: For  $x \in U$  we consider the matrix-valued map  $x \mapsto M(x) \in \mathbb{R}^{n \times n}$  and define  $F(x) := -M(x)^{-1}f(x)$ . Supposing that  $M(x)$  is invertible on a suitable subset of  $U$ , we now concentrate on the initial value problem

$$\begin{cases} \dot{x}(t) = F(x(t)), & t \geq 0, \\ x(0) = x_0. \end{cases} \quad (2)$$

This initial value problem tackles the problem of finding the zeros of  $f$  from a dynamical system approach. In fact, if  $M(x)$  is given by the Jacobian of  $f$  we recover the well known continuous Newton scheme formally satisfying  $f(x(t)) = f(x_0)e^{-t}$ . For an excellent survey of the continuous Newton scheme see, e.g., [8, 14–16]. Indeed, supposing that a solution  $x(t)$  exists for all time  $t \geq 0$ , we can try to follow the trajectory of  $x(t)$  numerically in order to end up with an approximate root for  $f$ . For an initial guess  $x_0 \in U$  the simplest routine for solving (2) numerically is given by the forward Euler method:

$$x_{n+1} = x_n - t_n M(x_n)^{-1} f(x_n), \quad t_n \in (0, 1], n \geq 0. \quad (3)$$

For example, if we choose  $M(x) := \text{Id}$ , the above iteration scheme is termed *Piccard-Iteration*. If  $J_f(x)$  signifies the Jacobian of  $f$  at  $x \in U$ , then for  $M(x) = J_f(x)$  we observe a *damped Newton-method*. Specifically, for  $t_n \equiv 1$  and  $M(x) = J_f(x)$ , the iteration (3) results in the well known *standard Newton method*. In addition, another well established scheme is given by setting  $M(x) := J_f(x_0)$ , which is also called *simplified Newton method*. The last choice simply freezes the information of the Jacobian throughout the whole iteration procedure. This typically reduces the computational effort in each iteration step. On the other hand, the number of iterations increases in general and the domain of convergence is reduced by this method. However, on a local level, i.e., when the initial guess  $x_0$  is supposed to be ‘sufficiently’ close to a zero of  $f$ , it is reasonable to expect that the simplified Newton method safely leads to a zero which is located next to the initial guess  $x_0$ . Indeed, if the update  $M(x_n)^{-1}f(x_n)$  is small enough, we will see in Sect. 2 that there exists a unique zero for  $f$  locally that can be obtained by the following simplified Newton-type iteration scheme:

$$u_{j+1} = u_j - M(x_n)^{-1} f(u_j), \quad u_n = x_n, \quad j \geq n. \quad (4)$$

This observation is especially interesting when the computation of the matrix  $M(x_n)$  is computationally expensive—as for instance when we solve extremely large scale non-linear problems arising from the discretization of PDE’s. Furthermore, the proposed result in this work asserts local uniqueness of the solution. Thus, one can think of steering an initial guess  $x_0 \in U$  assumed to be far away of a zero for  $f$ , ‘sufficiently’ close to the root which is located next to  $x_0$ . Having hit the domain of local uniqueness

of the underlying zero we then switch from the adaptive iteration (3) to the simplified iteration scheme given in (4) without using any adaptive step-size control.

## Notation

In this note we signify by  $(\cdot, \cdot)$  the standard Euclidean product of  $\mathbb{R}^n$ . For any  $x$  its norm is given by  $\|x\| := \sqrt{(x, x)}$ . For a matrix  $M \in \mathbb{R}^{n \times n}$  we further use the operator norm  $\|M\| := \sup_{\|x\|=1} \|Mx\|$ . By  $B_R(x)$  we denote the closed ball of radius  $R$  centered at  $x \in \mathbb{R}^n$ . Finally, whenever the function  $f$  is differentiable, the derivative at a point  $x \in U$  is written as  $J_f(x)$ , thereby referring to the Jacobian of  $f$  at  $x$ .

## Outline

This note is organized as follows: In Sect. 2 we state and prove a convergence result for a general class of simplified Newton-type iterations schemes as given in (4). Therefore we firstly discuss the assumptions that have to hold true in order to establish the proposed convergence result. In particular, we embed the local convergence result into a global—and therefore adaptive—Newton-type iteration scheme as given in (3). On that account, in Sect. 3 we finally present and discuss our adaptive strategy on a low dimensional example employing the advantage of the proposed iteration scheme. In Sect. 4 we summarize and comment our findings.

## 2 A convergence result

As a preparation towards the proposed main result we firstly address the assumptions that have to hold. In addition, we comment on a possible extension of the proposed result to a general Banach space framework.

### 2.1 Assumptions

Suppose we are given an initial value  $x_0 \in U$  and suppose we can compute

$$x_{j+1} = x_j - t_j M(x_j)^{-1} f(x_j), \quad t_j \in (0, 1], \quad j \geq 0. \quad (5)$$

Here,  $t_j$  signifies some adaptively chosen step size (see, e.g., [1,3,18,19] for some highly efficient step size methodologies). Again, we notice that for  $M(x) = J_f(x)$  and  $t_j \equiv 1$ , the iteration (5) is simply the *standard Newton method*.

Let  $U$  be an open and convex subset of  $\mathbb{R}^n$  and assume further that there exists an iterate  $x_n \in U$  such that there holds the following assumptions:

- A1.** Let  $\omega$  be a positive constant. For any  $v \in U$  and for any  $z \in \{tx_n + (1-t)v | t \in [0, 1]\}$  we assume that there holds the following affine covariant type Lipschitz-condition on  $J_f$ :

$$\left\| M(x_n)^{-1} (J_f(x_n) - J_f(z))(x_n - v) \right\| \leq \omega(1 - t) \|x_n - v\|^2. \quad (6)$$

**A2.** We further need  $M(x_n)^{-1}$  to be a sufficiently accurate approximate of the inverse of the Jacobian  $J_f(x_n)$  which we here quantify by the following assumption

$$\left\| \text{Id} - M(x_n)^{-1} J_f(x_n) \right\| \leq \kappa < 1.$$

**A3.** For  $\alpha_n := \|M(x_n)^{-1} f(x_n)\|$  we need to assume that

$$\omega \alpha_n \leq \frac{(1 - \kappa)^2}{2}. \quad (7)$$

**A4.** For

$$R := \frac{1 - \kappa}{\omega} + \sqrt{\frac{(1 - \kappa)^2}{\omega^2} - \frac{2\alpha_n}{\omega}} \quad (8)$$

there holds  $B_R(x_n) \subset U$ .

Assumption **A1** is called affine covariant type Lipschitz condition because in case of  $M(x_n) = J_f(x_n)$  the Lipschitz constant  $\omega$  is an affine invariant quantity. Indeed, for  $A \in \text{Gl}(n)$  and  $F(x) := Af(x)$  there holds

$$J_F(x_n)^{-1} (J_F(z) - J_F(x_n))(v - x_n) = J_f(x_n)^{-1} (J_f(z) - J_f(x_n))(v - x_n).$$

For further details concerning affine invariance principles within the framework of Newton-type iterations schemes we refer to the excellent monograph [8] and the proposed adaptive schemes therein.

Supposing that  $M(x_n)^{-1}$  is bounded, then condition (7) in **A3** also holds true whenever the residual  $\|f(x_n)\|$  is ‘sufficiently’ small in the sense that

$$\omega \alpha_n \leq \omega \left\| M(x_n)^{-1} \right\| \|f(x_n)\| \leq \frac{(1 - \kappa)^2}{2}. \quad (9)$$

Thus, the proposed result implies that whenever the norm of the residual  $\|f(x_n)\|$  is small enough, there exists a zero on a local level. This is of particular interest when solving nonlinear differential equations numerically within the context of a fully adaptive iteration scheme. More precisely, let  $X$  denote a Banach space—in most cases  $X = H_0^1(\Omega)$ —and  $X'$  its dual respectively. Then the weak formulation of a nonlinear differential equation reads as follows:

Find  $x \in X$  such that there holds

$$\langle f(x), v \rangle_{X' \times X} = 0 \quad \forall v \in X, \text{ i.e. } f(x) = 0 \text{ in } X', \quad (10)$$

with  $\langle \cdot, \cdot \rangle_{X' \times X}$  signifying the duality pairing in  $X' \times X$ .

Solving (10) within the context of an adaptive solution procedure over some finite dimensional space  $X_h \subset X$ —here  $h$  typically signifies the mesh-size parameter in the finite element method—one then can try to derive computational quantities  $\eta_{\mathbf{h}}(x_h)$  and  $\eta_{\mathbf{L}}(x_h)$  such that there holds:

$$\|f(x_h)\|_{X'} \leq \eta_{\mathbf{h}}(x_h) + \eta_{\mathbf{L}}(x_h). \quad (11)$$

Here, the quantity  $\eta_{\mathbf{L}}(x_h)$  signifies an error estimate which measures the linearization error whereas  $\eta_{\mathbf{h}}(x_h)$  represents the discretization error (see, e.g., [2,4–7,9–13]). Using (9) and supposing that the quantities  $\eta_{\mathbf{h}}(x_h)$  and  $\eta_{\mathbf{L}}(x_h)$  are small enough we obtain

$$\omega \left\| \mathbf{M}(x_n)^{-1} \right\|_{\mathcal{L}(X', X)} (\eta_{\mathbf{h}}(x_h) + \eta_{\mathbf{L}}(x_h)) \leq \frac{(1 - \kappa)^2}{2},$$

i.e., the a posteriori existence of the solution is guaranteed. Indeed, the a posteriori existence in numerical computations has been addressed in detail by [17]—especially in the context of solving semilinear problems. However, although we discuss and present our adaptive scheme in view of dealing with systems of nonlinear equations over  $\mathbb{R}^n$ , it is noteworthy that the established convergence result also holds true within a general Banach space setting. Indeed, our convergence result can be used to realize a specialization of the recently established adaptive iterative linearized Galerkin methodology (ILG) discussed in [12,13].

**Theorem 1** Suppose that  $f \in C^1(U; \mathbb{R}^n)$ . Further assume that there holds the assumptions A1&A2&A3 and A4.

Then the map

$$U \ni v \mapsto g(v) := v - \mathbf{M}(x_n)^{-1} f(v). \quad (12)$$

satisfies

$$g(B_R(x_n)) \subset B_R(x_n).$$

**Proof** First of all we rewrite the function  $g$  as follows

$$g(v) = x_n - \mathbf{M}(x_n)^{-1} f(x_n) - \left( (x_n - v) - \mathbf{M}(x_n)^{-1} (f(x_n) - f(v)) \right).$$

Let  $v \in B_R(x_n)$ . For  $t \in [0, 1]$  we define the line segment  $z(t) := tx_n + (1 - t)v \subset B_R(x_n)$  and use the integral form of the mean value theorem

$$\begin{aligned} (x_n - v) - \mathbf{M}(x_n)^{-1} (f(x_n) - f(v)) &= (x_n - v) - \int_0^1 \mathbf{M}(x_n)^{-1} \frac{d}{dt} f(z(t)) dt \\ &= \int_0^1 (\text{Id} - \mathbf{M}(x_n)^{-1} \mathbf{J}_f(z(t))) (x_n - v) dt \\ &= \int_0^1 \mathbf{M}(x_n)^{-1} (\mathbf{M}(x_n) - \mathbf{J}_f(z(t))) (x_n - v) dt \end{aligned}$$

$$\begin{aligned}
&= \int_0^1 (\text{Id} - M(x_n)^{-1} J_f(x_n))(x_n - v) dt \\
&\quad + \int_0^1 M(x_n)^{-1} (J_f(x_n) - J_f(z(t)))(x_n - v) dt
\end{aligned}$$

from where we obtain by **A1&A2**

$$\| (x_n - v) - M(x_n)^{-1} (f(x_n) - f(v)) \| \leq \left( \kappa + \omega \frac{1}{2} \|x_n - v\| \right) \|x_n - v\|. \quad (13)$$

Thus there holds

$$\|g(v) - x_n\| \leq \alpha_n + \left( \kappa + \omega \frac{1}{2} \|v - x_n\| \right) \|v - x_n\| \leq \alpha_n + \left( \kappa + \omega \frac{1}{2} R \right) R = R.$$

Employing **A3**, this last equality holds true if

$$R = \frac{1 - \kappa}{\omega} \pm \sqrt{\frac{(1 - \kappa)^2}{\omega^2} - \frac{2\alpha_n}{\omega}}. \quad (14)$$

□

Let us go back to (14) in the proof. We see that the map  $g$  also satisfies

$$g(B_r(x_n)) \subset B_r(x_n) \quad (15)$$

$$\text{with } r = \frac{1 - \kappa}{\omega} - \sqrt{\frac{(1 - \kappa)^2}{\omega^2} - \frac{\alpha_n}{\omega}}.$$

Next we give an existence result addressing the zeros  $u \in U$  of  $f$ .

**Corollary 2** *Assumptions and notations as in the preceding Theorem 1. Then, there exists a zero  $u \in B_R(x_n)$  of  $f$ .*

**Proof** From the proof of Theorem 1 we have that  $g(B_R(x_n)) \subset B_R(x_n)$ . Employing Brouwer's fixed point theorem we deduce the existence of a fixed point  $u \in B_R(x_n)$  of  $g$  which is the asserted zero of  $f$ . □

In view of the iteration procedure (4) it would be preferable if we can guarantee its convergence within the ball  $B_R(x_n) \subset U$ . Indeed, if  $g$  from (12) is a contraction in  $B_R(x_n)$  we can conclude the existence of a unique fixed point of  $g$  which can be obtained by iterating (4). In doing so we need to strengthen the assumptions **A1&A2&A3** and **A4** as follows:

**B1.** Let  $\omega^*$  be a positive constant. For any  $x, v \in U$  and for any  $z \in \{tx + (1-t)v | t \in [0, 1]\}$  we assume that there holds the following affine covariant type Lipschitz-condition on  $J_f$ :

$$\|M(x_n)^{-1} (J_f(x) - J_f(z))(x - v)\| \leq \omega^* (1 - t) \|x - v\|^2. \quad (16)$$

**B2.** For any  $x \in U$  there holds:

$$\left\| \text{Id} - M(x_n)^{-1} J_f(x) \right\| \leq \kappa^* < 1. \quad (17)$$

**B3.** For  $\alpha_n = \|M(x_n)^{-1} f(x_n)\| > 0$  we need to assume that

$$\omega^* \alpha_n \leq \frac{(1 - \kappa^*)^2}{2}. \quad (18)$$

**B4.** For

$$R^* := \frac{1 - \kappa^*}{\omega^*} + \sqrt{\frac{(1 - \kappa^*)^2}{\omega^{*2}} - \frac{2\alpha_n}{\omega^*}} \quad (19)$$

there holds  $B_{R^*}(x_n) \subset U$ .

Note that for  $x = x_n$  we have  $\omega = \omega^*$  and  $\kappa = \kappa^*$ . Now we are ready to prove the following result:

**Theorem 3** Suppose that  $f \in C^1(U; \mathbb{R}^n)$ . Further assume that there holds the assumption **B1&B2&B3&B4**. Then the map from (12) satisfies firstly

$$g(B_{R^*}(x_n)) \subset B_{R^*}(x_n)$$

and is a contraction on  $B_{R^*}(x_n)$ .

**Proof** The first assertion follows from the proof of Theorem 1 and choosing  $x = x_n$ . Thus we are left to show that  $g$  is a contraction. Notice that

$$\begin{aligned} g(x) - g(y) &= (x - y) - M(x_n)^{-1}(f(x) - f(y)) \\ &= \int_0^1 (\text{Id} - M(x_n)^{-1} J_f(z(t)))(x - y) dt \\ &= \int_0^1 (\text{Id} - M(x_n)^{-1} J_f(x))(x - y) dt \\ &\quad + \int_0^1 M(x_n)^{-1} (J_f(x) - J_f(z(t)))(x - y) dt. \end{aligned}$$

Thus, for  $x, y \in B_{R^*}(x_n)$  there holds:

$$\begin{aligned} \|g(x) - g(y)\| &= \left\| (x - y) - M(x_n)^{-1}(f(x) - f(y)) \right\| \\ &\leq \left( \kappa^* + \omega^* \frac{1}{2} \|x - y\| \right) \|x - y\|. \end{aligned}$$

Since  $\kappa^* + \omega^* \frac{1}{2} \|x - y\| \leq \kappa^* + \frac{\omega^*}{2} R^*$  and  $R^* < \frac{2(1-\kappa^*)}{\omega^*}$ , there holds

$$\kappa^* + \omega^* \frac{1}{2} \|x - y\| < 1,$$

i.e., we conclude that  $g$  is a contraction.  $\square$

**Corollary 4** *Assumptions and notations as in the preceding Theorem 3. Then, for any initial value  $x_n \in U$  the simplified Newton-like iterates (4) remain in  $B_{R^*}(x_n)$  and converge to a unique zero  $u_\infty \in B_{R^*}(x_n)$  of  $f$ .*

**Proof** From the proof of Theorem 1 we have that for  $j \geq n$  the iterates  $u_{j+1} = g(u_j)$  remain in  $B_{R^*}(x_n)$ . Furthermore we have also shown that  $g$  is a contraction on  $B_{R^*}(x_n)$ . Thus, by Banach's fixed-point theorem we deduce that  $\lim_{j \rightarrow \infty} g(u_j) = u_\infty$  exists, which is the unique zero of  $f$  in  $B_{R^*}(x_n)$ .  $\square$

From a computational point of view we can try to switch from the Newton-like iteration scheme (5) to a simplified Newton-like scheme

$$u_{j+1} = u_j - M(x_n)^{-1} f(u_j), \quad j \geq n \quad (20)$$

as soon as there holds  $\alpha_n \omega^* \leq \frac{(1-\kappa^*)^2}{2}$ . Therefore we need to control the Lipschitz constant  $\omega^*$ . In doing so, we replace the computational unavailable constant  $\omega^*$  by a quantity  $\hat{\omega}$  that we can easily compute during the iteration procedure. Henceforth, suppose we have computed  $x_{n+1}, x_n$ . In view of (6), it is reasonable to switch to the iteration (20) whenever there holds

$$\alpha_n \hat{\omega} = \alpha_n \frac{\|M(x_n)^{-1} (J_f(x_{n+1}) - J_f(x_n))(x_{n+1} - x_n)\|}{\|x_{n+1} - x_n\|^2} \leq \frac{(1-\kappa^*)^2}{2} < \frac{1}{2}.$$

In addition, for  $M(x_n) = J_f(x_n)$  and  $x \in B_{R^*}(x_n)$  we observe

$$\|\text{Id} - J_f(x_n)^{-1} J_f(x)\| \approx \|\text{Id} - J_f(x_n)^{-1} J_f(x_n)\| = 0,$$

i.e.  $\kappa = 0$ .

## 3 Numerical experiments

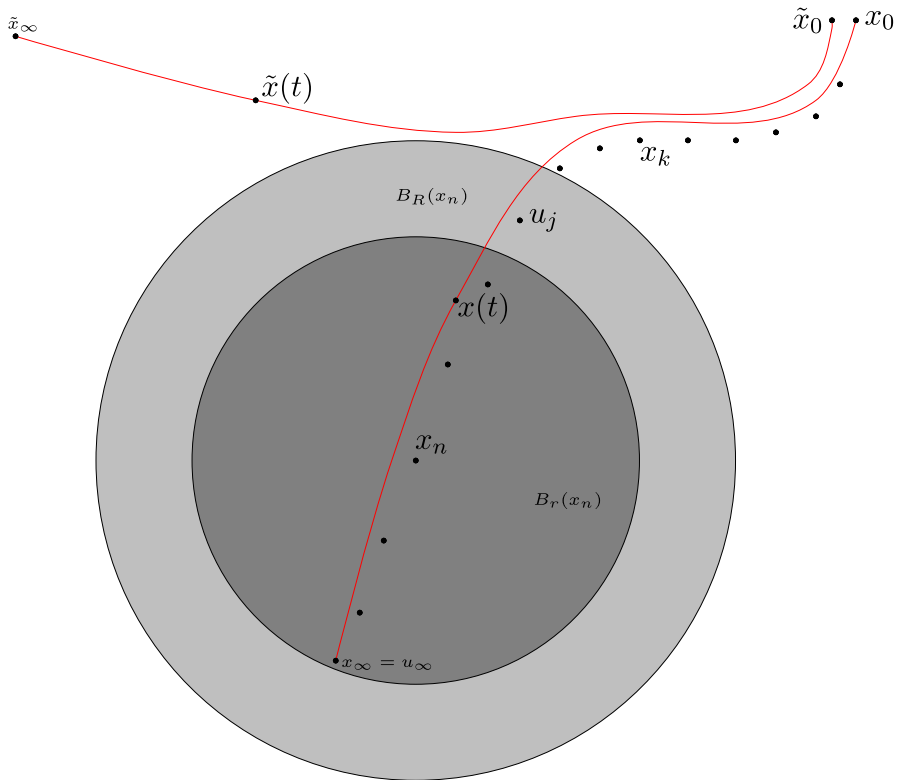
### 3.1 Adaptive strategy

We now propose a procedure that realizes an adaptive strategy based on the previous observations. The individual computational steps are summarized in Algorithm 1.

Let us briefly comment on the proposed adaptive procedure given in Algorithm 1:

**Remark 1** In steps 3&18 we predict a step size  $t$  such that  $t = 1$  whenever the iterates are 'close enough' to the zero  $x_\infty$ . Thus, the proposed procedure allows full steps





**Fig. 1** The adaptively computed sequence  $x_k$  switching to the simplified Newton-type scheme within the ball  $B_R(x_n)$  which finally leads to the zero  $x_\infty$ . Moreover, we depict two different trajectories  $x(t)$  and  $\tilde{x}(t)$  respectively—each of them leading to a different zero

whenever the iterates are ‘sufficiently’ close to  $x_\infty$ . The computation of  $t \in (0, 1]$  typically relies on a computational upper bound with respect to the distance  $\|x(t_n) - x_n\|$ . There exists different suggested approaches towards an effective computation of the step size  $t$  (see, e.g., [1,3,4,8,12,18,19]). Here we use the adaptive step size control given in [1]. This adaptive choice of the step size  $t$  consists mainly of two parts:

A prediction for the step size  $t$  and a correction of the step size whenever  $\|x_n - x(t_n)\| > \tau$ . Here,  $x(t)$  signifies the exact trajectory leading to the zero  $x_\infty$  and  $x_n$  is the numerical solution. Thus, the input  $\tau$  is a parameter that determines how close the iterates  $x_n$  tracks the exact trajectory  $x(t)$  leading to a zero of  $f$  (see also Fig. 1). For  $\tau = \infty$  there is no restriction on  $x_n$ , i.e. Algorithm 1 reproduces the classical Newton scheme—apart from the simplified Newton scheme given in step 13. Furthermore, the adaptive scheme from [1] needs a lower bound  $t_{\text{lower}}$  for the step size  $t_n$  in (3). Indeed, if  $t_n$  degenerates to 0, the iterative scheme is not well defined in the sense that it must be classified as not convergent. However,  $\tau$  is an error tolerance used in the proposed adaptive computation of the step size  $t$  and determines the distance between the numerically computed iterates and the exact trajectory.

**Algorithm 1** Adaptive simplified-Newton-like method:

---

```

1: Input:
    • initial value  $x_0 \in U$ ,
    • error tolerance  $\varepsilon > 0$  respectively.

2:  $\delta_0 \leftarrow -M(x_0)^{-1} f(x_0)$  ▷ compute a first correction
3:  $t \leftarrow \min(1, t)$  ▷ compute an initial step size based on an adaptive procedure; see, e.g., [1,18,19]
4:  $x_s \leftarrow x_0$ 
5: for  $k = 1, 2, \dots$  do
6:   if  $\|\delta_0\| \leq \varepsilon$  then
7:     return  $x_\infty \leftarrow x_0$  ▷ return the solution
8:   else
9:      $t \leftarrow t$  ▷ compute a step size based on an adaptive procedure; see, e.g., [1,18,19]
10:     $x_0 \leftarrow x_0 + t\delta_0$  ▷ perform a step
11:     $\hat{\omega} \leftarrow \frac{\|M(x_s)^{-1}(J_f(x_0) - J_f(x_s))(x_0 - x_s)\|}{\|x_0 - x_s\|^2}$  ▷ compute the Lipschitz constant
12:    if  $\|\delta_0\| \hat{\omega} \leq \frac{1}{2}$  then ▷ start the simplified Newton-like scheme
13:      Compute  $x_\infty$  based on the simplified iteration scheme (4)
14:      return  $x_\infty$  ▷ return the solution
15:      break the iteration
16:    end if
17:     $\delta_0 \leftarrow -M(x_0)^{-1} f(x_0)$  ▷ update the direction
18:     $t \leftarrow \min(1, t)$  ▷ predict the step size
19:     $x_s \leftarrow x_0$ 
20:  end if
21: end for

```

---

**Remark 2** Before we can switch to the *simplified Newton-type* iteration scheme in step 12, we need to compute the proposed estimate  $\hat{\omega}$  for the Lipschitz constant in step 11. Evidently, this will roughly increase the complexity of Algorithm 1 by a constant factor of 2—before we switch to the simplified scheme, where we can reduce the computational effort by keeping  $M(x_0)$  fixed for the rest of the iteration procedure—.

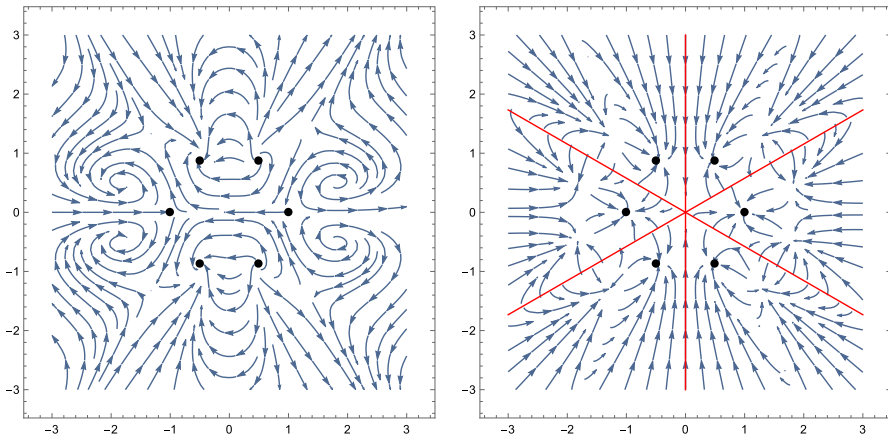
**Example 1** In this example we choose  $M(x) = J_f(x)$ . Let us consider the function

$$f : \mathbb{C} \rightarrow \mathbb{C}, \quad z \mapsto f(z) := z^6 - 1.$$

Here, we identify  $f$  in its real form in  $\mathbb{R}^2$ , i.e., we separate the real and imaginary parts. The six zeros are given by

$$Z_f = \{(1, 0), (1/2, \sqrt{3}/2), (-1/2, \sqrt{3}/2), (-1, 0), (-1/2, -\sqrt{3}/2), (1/2, -\sqrt{3}/2)\} \subset \mathbb{R}^2.$$

Note that  $J_f$  is singular at  $(0, 0)$ . Thus if we apply the classical Newton method with  $F(x) = -J_f(x)^{-1} f(x)$  in (3), the iterates close to  $(0, 0)$  cause large updates in the iteration procedure. More precisely, the application of  $F(x) = -J_f(x)^{-1} f(x)$  is a potential source for chaos near  $(0, 0)$ . Before we discuss our numerical experiment, let us first consider the vector fields generated by the continuous problem (2). In Fig. 2, we depict the direction fields corresponding to  $F(x) = f(x)$  (left) and  $F(x) = -J_f(x)^{-1} f(x)$  (right). We clearly see that some elements of  $Z_f$  are repulsive for  $F(x) = f(x)$ . Moreover, some elements of  $Z_f$  show a curl. If we now consider  $F(x) = -J_f(x)^{-1} f(x)$  the



**Fig. 2** Example 1: the direction fields corresponding to  $f(z) = z^6 - 1$  (left) and to the transformed  $F(z) = -J_f(z)^{-1} \cdot f(z)$  (right)

situation is completely different: All zeros are obviously attractive. In this example, we further observe that the vector direction field is divided into six different sectors, each containing exactly one element of  $Z_f$ .

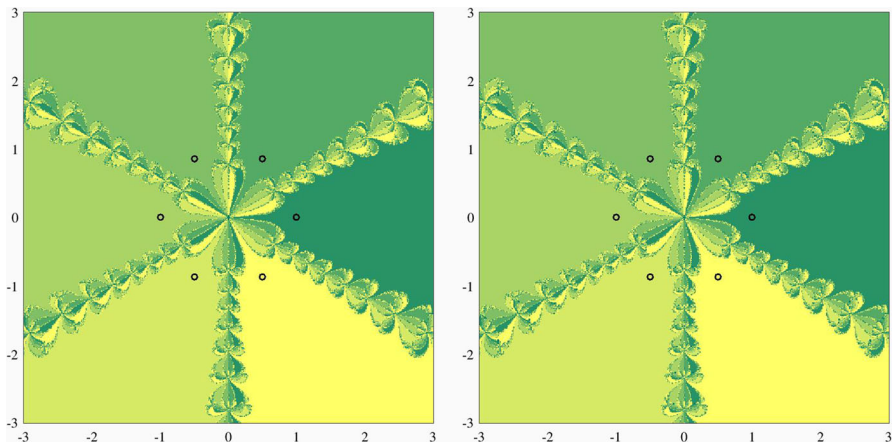
Next we visualize the domains of attraction of four different Newton-type iteration schemes. More precisely, we test the following four iteration procedures:

1. The proposed procedure given in Algorithm 1, i.e., adaptive step size control—with  $\tau = 0.01$ —and switching to the simplified Newton scheme which we abbreviate by **AS**.
2. The proposed procedure given in Algorithm 1, i.e., adaptive step size control—with  $\tau = 0.01$ —but without switching to the simplified Newton scheme which we abbreviate by **ANS**.
3. The proposed procedure given in Algorithm 1, without step size control, i.e.  $\tau = \infty$ , and without switching to the simplified Newton scheme which we abbreviate by **NANS**. This is simply the classical Newton iteration scheme.
4. The proposed procedure given in Algorithm 1, without step size control—i.e.  $\tau = \infty$ —but switching to the simplified Newton scheme which we abbreviate by **NAS**.

In doing so, we compute the zeros of  $f$  by sampling initial values on a  $500 \times 500$  grid in the domain  $[-3, 3]^2$  (equally spaced). In Fig. 3, we show the fractal generated by the traditional Newton method **NANS** (left) as well as the corresponding plot for the combination of the classical Newton method and the simplified Newton method **NAS** (right). It is noteworthy that the chaotic behavior caused by the singularities of  $J_f$  of the iteration procedure **NAS** is comparable to **NANS**.

In Fig. 4 we depict the basins of attraction for the adaptive procedure as proposed in Algorithm 1 **AS** (left) and the iteration procedure **ANS**. The chaotic behavior caused by the singularities of  $J_f$  is clearly tamed—by both adaptive schemes **AS** and **ANS**.

Let us finally consider some performance data given in Table 1. An initial value  $x_0 \in [-3, 3]^2$  is called convergent if it is in fact convergent and additionally approaches the ‘correct’ zero, i.e., the zero that is located in the same exact attractor as the initial



**Fig. 3** The basins of attraction for Example 1 by the Newton method. On the left for the classical Newton scheme **NANS** and on the right using the proposed simplified Newton iteration scheme **NAS**. Different colors distinguish the six basins of attraction associated with the six solutions (each of them is marked by a small circle). (Color figure online)

**Table 1** Performance for Examples 1

	AS	ANS	NANS	NAS
Convergent	99.98%	99.98%	80.4%	80.4%
Complexity	2.25	2.7	1	1.03

Here we clearly see the advantage of the proposed adaptive procedure based on the simplified Newton-type scheme **AS**. This is due to fixed derivative  $M(x_n) = J_f(x_n)$  as soon as  $\alpha_n \omega \leq 1/2$ . Furthermore, we see that almost all tested initial guesses are converging to the correct zero

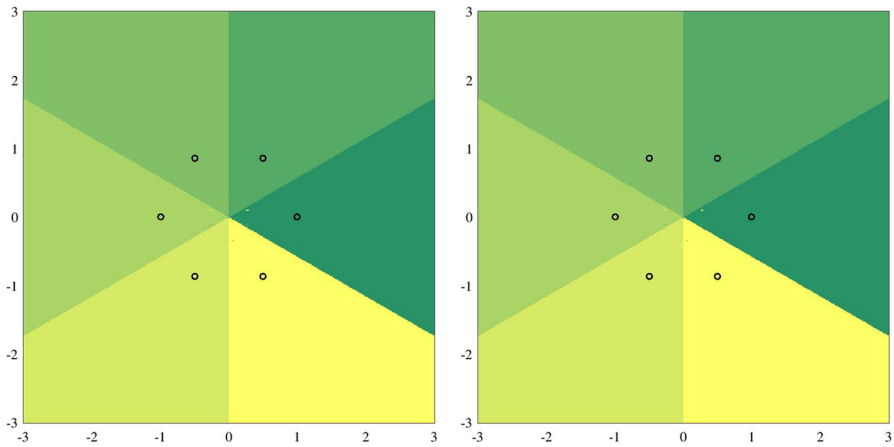
guess  $x_0$ . Table 1 nicely demonstrates that—in contrast to the non adaptive schemes **NANS** and **NAS**—the number of convergent iterations for the adaptive procedures **AS** and **ANS** is close to 100%. The second line in Table 1 shows the computational time—by sampling the computational time for all tested initial guesses  $x_0 \in [-3, 3]^2$ —with respect to the classical Newton iteration scheme **NANS**, i.e., we depict the quantity

$$\frac{\text{Computational time of the considered iteration scheme}}{\text{Computational time of NANS}}.$$

In view of this quantity, the proposed iteration scheme **AS** is the clear winner compared to **ANS** as can be seen from line 2 in Table 1.

## 4 Conclusions

In this work, we have proved a convergence result for general simplified Newton-type iteration schemes under quite reasonable assumptions. In particular, we have shown



**Fig. 4** The basins of attraction for Example 1 by the Newton method. On the left with step size control (i.e.,  $t \in (0, 1]$ ) and the proposed scheme based on Algorithm 1 AS. On the right again with step size control (i.e.,  $t \in (0, 1]$ ) but without the simplified scheme, i.e., the derivative  $J_f$  was updated in each iteration step ANS. Six different colors distinguish the six basins of attraction associated with the six solutions (each of them is marked by a small circle). (Color figure online)

that whenever the correction  $\|M(x_n)^{-1}f(x_n)\|$  is small, then there locally exists a unique zero for the underlying map  $f$ . Since the proof of the proposed result relies on Banach's fixed-point theorem, the theoretical result is constructive in the sense that it can be used for the numerical computation of the locally unique fixed point and therefore of the zero to be considered. Moreover, we have combined the convergence result with an adaptive root finding procedure thereby firstly taming the chaotic behavior of classical Newton-type iteration schemes and secondly reducing the computational effort due to the constant map  $x \mapsto M(x_n) \in \mathbb{R}^{n \times n}$ —without reducing the domain of convergence. We have tested our method on a low dimensional problem. Moreover, our experiment demonstrates empirically that the proposed scheme is indeed capable to tame the *chaotic* behavior of the iteration compared with the classical Newton scheme, i.e., without applying any step size control. In particular, our test example illustrate that the domains of convergence can—typically—be considerably enlarged in the sense that almost all initial guesses  $x_0$  are convergent to the ‘correct’ zeros, i.e., the zero which is located in the same attractor as the initial guess  $x_0$ .

**Acknowledgements** Open access funding provided by ZHAW Zurich University of Applied Sciences. The author is grateful to Pascal Heid for comments on an earlier draft of this manuscript.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Amrein, M.: Adaptive Newton-Type Schemes Based on Projections, Tech. Report. [arXiv:1809.04337v2](https://arxiv.org/abs/1809.04337v2) (2018)
2. Amrein, M., Melenk, J.M., Wihler, T.P.: An hp-adaptive Newton–Galerkin finite element procedure for semilinear boundary value problems. *Math. Methods Appl. Sci.* **40**(6), 1973–1985 (2016). 13 pages, mma.4113
3. Amrein, M., Wihler, T.P.: An adaptive Newton-method based on a dynamical systems approach. *Commun. Nonlinear Sci. Numer. Simul.* **19**(9), 2958–2973 (2014)
4. Amrein, M., Wihler, T.P.: Fully adaptive Newton–Galerkin methods for semilinear elliptic partial differential equations. *SIAM, J. Sci. Comput.* **37**(4), A1637–A1657 (2015). 21 pages
5. Chaillou, A., Suri, M.: A posteriori estimation of the linearization error for strongly monotone nonlinear operators. *J. Comput. Appl. Math.* **205**(1), 72–87 (2007)
6. Chaillou, A.L., Suri, M.: Computable error estimators for the approximation of nonlinear problems by linearized models. *Comput. Methods Appl. Mech. Eng.* **196**(1–3), 210–224 (2006)
7. Congreve, S., Wihler, T.P.: Iterative Galerkin discretizations for strongly monotone problems. *J. Comput. Appl. Math.* **311**, 457–472 (2017)
8. Deufhard, P.: *Newton Methods for Nonlinear Problems*, Springer Series in Computational Mathematics. Springer, Berlin (2004)
9. Dörfler, W.: A robust adaptive strategy for the nonlinear Poisson equation. *Computing* **55**(4), 289–304 (1995)
10. El Alaoui, L., Ern, A., Vohralík, M.: Guaranteed and robust a posteriori error estimates and balancing discretization and linearization errors for monotone nonlinear problems. *Comput. Methods Appl. Mech. Eng.* **200**(37–40), 2782–2795 (2011)
11. Han, W.: A posteriori error analysis for linearization of nonlinear elliptic problems and their discretizations. *Math. Methods Appl. Sci.* **17**(7), 487–508 (1994)
12. Heid, P., Wihler, T.P.: Adaptive Iterative Linearization Galerkin Methods for Nonlinear Problems, Tech. Report. [arXiv:1808.04990](https://arxiv.org/abs/1808.04990) (2018)
13. Heid, P., Wihler, T.P.: On the Convergence of Adaptive Iterative Linearized Galerkin Methods, Tech. Report. [arXiv:1905.06682](https://arxiv.org/abs/1905.06682) (2019)
14. Neuberger, J.W.: Continuous Newton’s method for polynomials. *Math. Intel.* **21**, 18–23 (1999)
15. Neuberger, J.W.: Integrated form of continuous Newton’s method. *Lect. Notes Pure Applied. Math.* **234**, 331–336 (2003)
16. Neuberger, J.W.: The continuous Newton’s method, inverse functions and Nash Moser. *Am. Math. Mon.* **114**, 432–437 (2007)
17. Ortner, C.: A posteriori existence in numerical computations. *SIAM J. Numer. Anal.* **47**(4), 2550–2577 (2009)
18. Potschka, A.: Backward step control for global Newton-type methods. *SIAM J. Numer. Anal.* **54**(1), 361–387 (2016)
19. Schneebeli, H.R., Wihler, T.P.: The Newton–Raphson method and adaptive ODE solvers. *Fractals* **19**(1), 87–99 (2011)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.